Los Alamos NATIONAL LABORATORY EST.1943

LA-UR-15-21231 (Accepted Manuscript)

dfnWorks: A discrete fracture network framework for modeling subsurface flow and transport

Hyman, Jeffrey De'Haven
Karra, Satish
Makedonska, Nataliia
Gable, Carl Walter
Viswanathan, Hari S.
Painter, Scott L.

Provided by the author(s) and the Los Alamos National Laboratory (2016-05-09).

To be published in: Computers & Geosciences

DOI to publisher's version: 10.1016/j.cageo.2015.08.001

Permalink to record: http://permalink.lanl.gov/object/view?what=info:lanl-repo/lareport/LA-UR-15-21231

Disclaimer:
Approved for public release. Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the Los Alamos National Security, LLC for the National Nuclear Security Administration of the U.S. Department of Energy under contract DE-AC52-06NA25396. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

RESEARCH LIBRARY

dfnWorks: A discrete fracture network framework for modeling subsurface flow and transport

Jeffrey D. Hyman[a,b,*], Satish Karra[a], Nataliia Makedonska[a], Carl W. Gable[a], Scott L. Painter[c], Hari S. Viswanathan[a]

[a] *Computational Earth Science Group, Earth and Environmental Sciences Division, Los Alamos National Laboratory, Los Alamos, NM 87545, United States of America*
[b] *Center for Nonlinear Studies, Los Alamos National Laboratory, Los Alamos, NM 87545, United States of America*
[c] *Environmental Sciences Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831, United States of America*

Abstract

dfnWorks is a parallelized computational suite to generate three-dimensional discrete fracture networks (DFN) and simulate flow and transport. Developed at Los Alamos National Laboratory over the past five years, it has been used to study flow and transport in fractured media at scales ranging from millimeters to kilometers. The networks are created and meshed using dfnGen, which combines fram (the feature rejection algorithm for meshing) methodology to stochastically generate three-dimensional DFNs with the LaGriT meshing toolbox to create a high-quality computational mesh representation. The representation produces a conforming Delaunay triangulation suitable for high performance computing finite volume solvers in an intrinsically parallel fashion. Flow through the network is simulated in dfnFlow, which utilizes the massively parallel subsurface flow and reactive transport finite volume code pflotran. A Lagrangian approach to simulating transport through the DFN is adopted within dfnTrans to determine pathlines and solute transport through the DFN. Example applications of this suite in the areas of nuclear waste repository science, hydraulic fracturing and CO_2 sequestration are also included.

Keywords: discrete fracture networks, conforming Delaunay triangulation, subsurface flow and transport, fractured porous media, hydraulic fracturing, CO_2 sequestration

1. Introduction

Discrete fracture network (DFN) modeling is an alternative to continuum approaches for simulating flow and transport through sparsely fractured rocks in the subsurface. In contrast to continuum methodologies, e.g., stochastic continuum [1] and dual/multiple continua [2], where effective parameters are used to include the influence of the fractures on the flow, in the DFN approach, geologic field investigations are used to create a network of fractures where the geometry and properties of individual fractures are explicitly represented as lines in two dimensions or planar polygons in three dimensions. These generated networks are meshed for computation and the governing equations are numerically integrated to simulate flow. Examples of the various DFN methodologies and their applications are found in [3, 4, 5, 6, 7, 8, 9, 10, 11, 12].

A primary challenge in using the DFN methodology is creating an efficient and scalable workflow. The choice to include the detailed geometry of the fractures and the connectivity of the fracture network allows for a more accurate representations of physical phenomenon and robust predictive simulation of flow and transport through fractured rocks compared to continuum approaches. However, these advantages come at an enormous computational cost, especially when attempting to simulate transport through large networks

*Corresponding Author: Jeffrey Hyman email: jhyman@lanl.gov

of fractures. This challenge can be broken down into three main obstacles: i) generation of a high quality computational mesh representation of three-dimensional fracture network, ii) solving the governing equations on that mesh in a computationally efficient manner, and iii) simulating transport through the resulting flow field.

In this paper, we describe the dfnWorks high-performance computational suite that overcomes each of these obstacles in a unique fashion. Developed at Los Alamos National Laboratory over the past five years, dfnWorks provides a novel workflow to model flow and transport in three-dimensional fractured media at scales ranging from millimeters to kilometers[1]. Figure 1 illustrates the utility of the dfnWorks workflow with a simple two fracture network. Fractures are colored by the steady-state pressure solution, the gradient is aligned with the x-axis, and black lines are particle trajectories. A pipe-network representation that preserves the topology of the DFN while disregarding the detailed geometry of the fractures, a common practice in many DFN simulations, e.g., [3, 13, 14], would eliminate this dead-end fracture from the network, thereby disregarding the influence of flow in the elliptical fracture on transport. This figures shows that the influence of flow in this fracture on transport is non-negligible and how the simplifying assumption misses important transport properties. The non-uniformity of the trajectories results in a distribution of particle travel times that exhibits longitudinal dispersion. Using the simplifying pipe-network approximations, the breakthrough curve would be a step function with no dispersion or tail, which will lead to incorrect upscaled models for transport. Although this is a sample two fracture example, it serves to demonstrate the importance of both the geometry and topology of these networks.

In Sec. 2 we describe each of these pillars of dfnWorks (dfnGen, dfnFlow, dfnTrans) which make simulations similar to Fig. 1 feasible. In Sec. 3 we provide three example DFN simulations to highlight various features of the computational suite and show its suitability for real world applications. We conclude with some remarks about the suite and discuss some extensions of the method in Sec. 4.

2. dfnWorks Description

An overview of the entire dfnWorks workflow is illustrated in Fig. 2. The workflow has three principal pieces (dfnGen, dfnFlow, dfnTrans) which can be broken down into six primary aspects. The input for dfnWorks is a fractured site characterization that provides distributions of fracture orientations, radius, and spatial locations. dfnGen: 1) fram - Create DFN: Using the fractured site characterization networks are constructed using the feature rejection algorithm for meshing. 2) LaGriT - Mesh DFN: The LaGriT meshing tool box is used to create a conforming Delaunay triangulation of the network. dfnFlow: 3) Convert Mesh to pflotran input: Control volume information is formatted for pflotran. 4) Compute Pressure Solution: The steady-state pressure solution in the DFN is obtained using pflotran. dfnTrans: 5) Reconstruct Local Velocity Field: Darcy fluxes obtained using dfnFlow are used to reconstruct the local velocity field, which is used for particle tracking on the DFN. 6) Lagrangian Transport Simulation: An extension of the walkabout method [15] is used to determine pathlines through the network and simulate transport. It is important to note that dfnTrans itself only solves for advective transport, but effects of longitudinal dispersion and matrix diffusion, sorption, and other retention processes are easily incorporated by post-processing particle trajectories [16, 15]. Communication between the different pieces of the dfnWorks workflow is carried out using files, which also allows for restarts between the different modules of the code. Various python scripts are used to format the output from one piece of the workflow into the required input format of the next piece. All coupling between the pieces of the workflow is fully automated and does not require user actions. One of the key features of dfnWorks is that it combines existing software, e.g., LaGriT and pflotran, in a novel workflow. The primary benefits of this choice to aggregate codes that have already been optimized is efficiency, longevity, and that verification and validation have already been performed by independent parties.

In the rest of this section, we describe the different pieces of the dfnWorks framework in the order that they are implemented during the workflow.

[1] dfnWorks is open source, released under LA-CC #14-091 and can be obtained by contacting dfnworks@lanl.gov

2.1. dfnGen: *The Feature Rejection Algorithm for Meshing*

Each three-dimensional DFN is generated and meshed using the feature rejection algorithm for meshing (fram) methodology of Hyman et al., [17]. Each DFN is constructed so that all features in the network, e.g., length of intersections between fractures and distance between lines of intersection of a fracture are larger than a user-defined minimum length scale. This restriction provides a firm lower bound on the required mesh resolution, and special care is taken so that prescribed geological statistics are not affected by this restriction. Once the DFN is generated, the LaGriT [18] meshing toolbox is used to create a high resolution computational mesh representation of the DFN in parallel. An algorithm for conforming Delaunay triangulation is implemented so that fracture intersections are coincident with triangle edges in the mesh and Voronoi control volumes suitable for finite volume flow solvers such as fehm [19], tough2 [20], and the massively parallel pflotran [2] are produced. Because the mesh conforms to fracture intersections, the method does not require solving additional systems of linear equations, which is needed if a non-conforming mesh is used [4, 10, 11]. Key features of fram are that it is: a) fully automated – meaning that throughout the procedure no adjustments of the mesh are performed to improve the mesh quality; and b) it is flexible – any statistical survey of a fracture site may be used in the generation of fractures, thereby allowing it to generate realistic DFNs that mimic natural fracture sites. The generation portion of dfnGEN, fram , is written in C++ and LaGriT is written in Fortran. In this section, we provide a brief description of fram and refer the interested reader to Hyman et al., [17] for a complete details of the method.

2.1.1. *Network Generation and Meshing*

The principal issue in meshing a DFN is that to resolve a tiny feature in the network, the edges of the mesh surrounding the feature must be the size of the feature or smaller, if the physics are to be properly resolved. Various methods have been proposed to address this issue, and others, associated with meshing a DFN. In one methodology, pathological cases that degrade mesh quality, e.g., an arbitrarily short line of intersection between two fractures, are systematically removed after an unconstrained fracture network is generated and meshed [21, 8, 9, 22, 23, 24]. However, such adjustments can deform the network, resulting in fractures that may no longer be planar. Another methodology that gaining popularity is not requiring meshes to coincide along traces. In such a formulation the flow solution across intersecting fractures can be explicitly incorporated into the governing PDEs or computationally by using mortar methods [25, 4, 11, 10]. Their advantage is that they limit the features in network that the mesh must resolve because the mesh does not need to conform to the traces. In terms of the solver, these methods are computationally more expensive than when the meshes are forced to align due to additional interaction terms that must be computed. In addition, a non-aligned grid at fracture intersections may complicate solute transport calculations.

In contrast to these methods, dfnGen uses fram to constrain the generation of the network so that it only contains features greater than or equal to a user-prescribed minimum length scale h. Each fracture in our DFN is a planar straight-line graph (PSLG) made up of the set of line segments that represent the boundary of the fracture and the line segments that represent where other fractures intersect it. Given a set of PSLGs $\mathcal{X}$ with arbitrary orientation in $\mathbb{R}^3$, one can define a *local feature size at a point* p as the radius of the smallest sphere centered at p that intersects two non-incident vertices of segments of $\mathcal{X}$ [26]. In a three-dimensional DFN, examples of a measurable feature include: the length of the line of intersection between two fractures, the distance from the end of a fracture intersection that is interior to the polygon boundary to the polygon boundary, and the distance between two fracture intersection line segments. During the generation process, we require that the DFN never generates a fracture with a feature of size less than h, which provides a firm lower bound on the required resolution of the mesh. When the resulting network is meshed, all features can be resolved by generating triangular cell edges with a minimum length slightly less than h.

By constraining the network so that all features in the network are greater than h, fram ensures that pathological cases which degrade mesh quality are not present in the network. Under these conditions, a conforming Delaunay triangulation algorithm [27] can be used to ensure that the line of intersection between any two fractures is preserved in the mesh so long as the lines of intersection are discretized in steps less than h. The conforming Delaunay triangulation algorithm procedure results in meshes that are coincident

along the common line of intersection between fractures. Because computational control volumes (Voronoi polygons) are based on vertices and the triangular meshes are coincident along intersections, the Voronoi cells also conform at the fracture boundaries. This results in Voronoi control volumes that span both of the intersecting fractures. However, the neighbors of these Voronoi cells are still two-dimensional. The need to check for a feature size less than h means that the fracture generation process is computationally more demanding than methods that do not impose the minimum feature size constraint. The tradeoff is a streamlined/parallelized process of mesh generation, numerical integration of the pressure solution, and simplifying particle tracking through the resulting flow field.

In Fig. 3 three intersecting fractures show the intersecting conforming Delaunay triangulations. Two of the fractures are colored by distance from lines of intersections (traces) with each other and other fractures that intersect these fractures, and the other is semi-transparent for clarity. The mesh is optionally coarsened away from intersections with pressure gradients will be lower. The inclusion of the semi-transparent fracture illustrates how FRAM creates a mesh that adheres to multiple intersections on the surface of a single fractures. Two additional fractures intersect the elliptical fracture, and intersect one another on the surface of that fracture, as shown by the intersecting white colored regions; the initial version of FRAM did not allow for intersections of intersections, but the most current version, that used here, does. The inset shows that the Delaunay mesh conforms all of these lines of intersection.

Decisions about the minimum length scale h that will be represented in a DFN are made a priori which is typically the case in scientific computing and not unique to FRAM. When adopting the FRAM methodology the choice of the h will be reflected in the generated network. If h is chosen to large with respect to fracture and domain size, then it will be difficult to generate a DFN that meets the density requirements. If h is chosen too small, then computational cost associated with meshing and solving the governing equations will increase. The choice of h should be made so that all physical phenomena of interest have natural scales greater than h, so they can be well resolved by the computational mesh, while limiting computational expenses. The tradeoff between spatial resolution and computational expediency inherent in the choice of h in FRAM is the familiar tradeoff in most branches of scientific computing.

Due to the rejection nature of FRAM some of the desired distributions in the network, e.g., fracture length, will not be properly represented due to over rejection unless certain criteria are met. For example, larger fractures generate more measurable features in the network than smaller ones and can be rejected disproportionately. However, modifying the procedure by which fracture lengths are sampled can alleviate the issue of bias in the represented fracture length distribution. This is only necessary when sampling from a distribution with a broad range of lengths, such as a truncated power law distribution. An alternative solution is decreasing h to loosen the acceptation criteria. In the limit of $h \to 0$, all prescribed distributions will be recovered exactly because no fractures are rejected. Details about these procedures are in [17].

2.2. DFNFLOW*: Flow Solver*

Once the DFN is generated and meshed using DFNGEN, Voronoi control volumes, the dual mesh of the Delaunay triangulation, are computed. Then a Python script processes the LAGRIT output into an unstructured grid format compatible with the massively parallel subsurface flow code PFLOTRAN. Modifications were made to PFLOTRAN to read the unstructured Voronoi mesh and to perform calculations on the Voronoi mesh. The format to read Voronoi meshes involves locations of the Voronoi cell-centers, the connectivity of the cell-centers and the cell areas, and is referred to as explicit unstructured grid format. The adopted explicit unstructured grid format can be used to read any generalized n-faced polygonal meshes. Using this information, PFLOTRAN determines the steady-state pressure field within the DFN using a two-point flux based finite volume scheme. By using a control volume based discretization for flow, local mass balance is ensured, and by using a Voronoi mesh, the accuracy of the flux evaluation is maintained [28]. PFLOTRAN returns the fluid fluxes (Darcy velocities) at the edges of the Voronoi control volumes.

The subsurface flow processes currently available in PFLOTRAN include single-phase variably-saturated flow (Richards equation), non-isothermal two-phase water-supercritical CO_2, general mode that allows modeling multiphase flow of any two component systems, thermal-hydrologic coupled heat and mass conservation and three-phase ice-liquid-vapor (water) flow for Arctic applications. Any of these flow modes can be coupled to multicomponent reactive transport equations using either a global implicit algorithm or operator

splitting. The multicomponent reactive transport processes include aqueous complexation, sorption, mineral precipitation/dissolution, and microbially-mediated biodegradation. Because these flow processes are already available in PFLOTRAN, they can all be used for DFN simulations with little extended effort on the user's part. In the next part of this section, a brief description of PFLOTRAN is presented. The reader is referred to [29, 30, 31] for details of PFLOTRAN implementation and its parallel performance.

PFLOTRAN [2] is an open-source, massively parallel, multiscale and multiphysics code for subsurface and surface processes. The code is the result of a multiple U.S. Department of Energy national laboratory effort with core developers from LANL, SNL, LBNL and ORNL, with users and contributors from universities and other research facilities all over the world. PFLOTRAN is built on top of PETSc (Portable, Extensible Toolkit for Scientific Computation) [32] framework and incorporates PETSc's parallel data structures, linear and non-linear solvers, and relies on domain decomposition for parallelism, and was originally developed as part of the Department of Energy SciDAC-2 groundwater program. This foundation has allowed PFLOTRAN to run simulations with billions of degrees of freedom on over 200,000 processor cores. The code solves a system of nonlinear partial differential equations that model non-isothermal multiphase flow, reactive transport and geomechanics in porous media. Equations describing multiple continua (for matrix-fracture interactions) and non-isothermal surface flow can also solved. The code is written in object-oriented Fortran 2003 (base and derived classes) which gives the flexibility to: a) add new process models and couple with existing process models, and b) couple with other external codes that use PETSc and drive the coupling with PFLOTRAN as the master. To reduce I/O bottlenecks, PFLOTRAN also has parallel I/O modules built and outputs data in Tecplot ASCII as well as the binary HDF5 formats. The HDF5 output can be visualized using the HPC visualization toolkits such as VisIt [33] and Paraview [34]. PFLOTRAN can run on a wide range of architectures ranging from multi-core laptops and clusters to leadership-class supercomputers. PFLOTRAN applications are related to areas in energy, climate and nuclear including nuclear waste disposal [35], CO_2 sequestration [36, 37], enhanced geothermal systems [36], groundwater contamination [38], hydraulic fracturing [39, 40], induced seismicity [41] and Arctic hydrology and climate [42, 43]. PFLOTRAN can be downloaded from http://bitbucket.org/pflotran.

2.3. DFNTRANS*: Particle Tracking*

DFNTRANS is a method for resolving solute transport using control volume flow solutions obtained from DFNFLOW on the unstructured mesh generated using DFNGEN. We adopt a Lagrangian approach and represent a non-reactive conservative solute as a collection of indivisible passive tracer particles. Particle tracking methods: a) provide a wealth of information about the local flow field, b) do not suffer from numerical dispersion, which is inherent in the discretizations of advection-dispersion equations, and c) allow for the computation of each particle trajectory to be performed in an intrinsically parallel fashion if particles are not allowed to interact with one another or the fracture network. However, particle tracking on a DFN poses unique challenges that arise from both the quality of the flow solution, the unstructured mesh representation of the DFN, and the physical phenomena of interest. The flow solutions obtained from DFNFLOW are locally mass conserving, so the particle tracking method does not suffer from the problems inherent in using Galerkin finite element codes, e.g., [7, 44, 45, 46, 47, 48], where experience has shown that particles can become stuck in cells that exhibit unphysical stagnant regions because the flow solution does not conserve mass locally. Research codes based on mixed hybrid finite elements, such as discontinuous Galerkin methods, are locally mass conserving and have recently grown in popularity [6, 24]. These methods presumably should produce velocity fields that are amenable for particle tracking on a DFN. Because the primary interest of particle tracking is simulation of mass transport, these stuck particles, representing a loss of mass in the system due to numerical artifacts, limit the usefulness of flow field provided by finite element methods [47]. The coupling of DFNTRANS with finite volume codes ensures that all particles released into these DFN will eventually exit the system, and do so without incorporating additional pipe-network simplifications [3, 13].

In this section, we highlight the most important details of the method, namely: (i) the reconstruction of velocities on each vertex of the computational mesh and (ii) the unique approach taken to address mass transport through fractures intersections. A comprehensive explanation of DFNTRANS and details concerning

its implementation can be found in Makedonska et al., [35]. DFNTRANS is written in object-oriented C for speed and detailed control of memory allocation.

2.3.1. Reconstruction of Velocity Field

The pathline followed by a particle is obtained by numerically integrating the trajectory equation, $\dot{\mathbf{x}} = \mathbf{v}(\mathbf{x}(t))$, with initial position $\mathbf{x}(0) = \mathbf{x}$. Here, $\mathbf{v}$ is the Eulerian velocity vector that must be defined at every point in the fracture network. However, the control volume flow solution provides a set of scalar quantities that are approximations to the normal component of Darcy flux integrated over each edge of each control volume cell, rather than the required continuous velocity field.

To address this issue Painter et al. [15] developed an approach to reconstruct velocity fields from flow solutions obtained on unstructured control volume mesh. Using the flow solution provided by DFNFLOW, cell-centered velocities are estimated by solving a overdetermined linear system for the Darcy velocity. The system is overdetermined because each control volume has more sides than the dimension of the problem, e.g., in two dimensions each control volume has a minimum of three sides. On cells in the interior of the domain an unconstrained least squares method is used, and a constrained least squares method used to reconstruct velocities on boundary cells to enforce Neumann boundary conditions along fracture walls. This procedure creates piecewise constant flow velocity vectors at every vertex in the DFN mesh. The particular method for reconstructing the velocity field was chosen because of its ease of implementation and its compatibility with our particular representation of cells at fracture intersections. Then barycentric interpolation [49] is used to determined a particle's velocity at any location within a cell at any point in the DFN. Using these velocities, an adaptive time stepping first-order predictor-corrector method is used to numerically integrate the trajectory equation. The semi-implicit nature of the predictor-corrector method prevents particles from reaching the edge of a fracture with no-flow boundary conditions.

2.3.2. Fracture Intersections

Simulating transport through fracture intersections is a principal challenge for modeling transport through a three-dimensional DFN. The control volume cells along fracture intersections are three-dimensional objects formed from the union of two polygons in different planes; control volume cells away from intersections are two-dimensional planar polygons. This peculiar shape at intersections results in more complicated flow behavior than elsewhere in the network. Flow can go through the intersection and continue onto the same fracture without changing direction, or it change direction and exit onto the intersecting fracture or, as is usually the case, the flow can split by some percentage between the fractures.

We model this phenomenon by assuming complete mixing occurs at fracture intersections, as opposed to streamline routing [50, 51, 52]. A flux-weighted stochastic method is used to determine how particles exit fracture intersections. The technique schematically presented in Fig. 4 was developed to provide the necessary information so that the number of particles dispersed at fractures intersections is representative of the percentage of flux outgoing onto each fracture at intersections. Figure 4.a shows a control volume along a line of intersection between two fractures. The control volume is a three-dimensional objects formed from the union of two polygons in different planes where the Darcy flux is defined along the boundaries of the control volumes. The first step is dividing the control volume into four sub-polygons using the line of intersection, this partition is shown in Fig. 4.b. Here, the green and orange sub-polygons are on one fracture and the blue and purple are on another. Then the flow velocities are reconstructed on each of the split polygons, where arrows indicate inflow/outflow. In the example shown in Fig. 4.c, the flux into the control volume occurs through the purple sub-volume and outflow occurs through the remaining sub-polygons; any combination of inflow/outflow can be accommodated. To determine how particles pass through intersections, probabilities proportional to the outgoing flux are assigned to each cell that borders the intersection. Then, the downstream cell is chosen randomly based on these probabilities. This method ensures that the percentage of particles exiting onto each cell is proportional to the flux exiting onto those cells.

3. Applications

In this section we provide three applications of DFNWORKS to demonstrate its utility in various subsurface scenarios. All of the example DFN can be obtained by emailing dfnworks@lanl.gov.

3.1. Nuclear Waste Repository

The Swedish Nuclear Fuel and Waste Management Company (SKB) has undertaken a detailed investigation of the fractured granite at the Forsmark, Sweden site as a potential host formation for a subsurface repository for spent nuclear fuel [53, 54]. The Forsmark area is about 120 km north of Stockholm in northern Uppland, and the repository is proposed to be constructed in crystalline bedrock at a depth of approximately 500 m. Based on the SKB site investigation, a statistical fracture model with multiple fracture sets was developed; detailed parameters of the Forsmark site model are in [53]. We adopt a subset of the model that consist of three sets of background (non-deterministic) circular fractures whose orientations follow a Fisher distribution, fracture radii are sampled from a truncated power-law distribution, the transmissivity of the fractures is estimated using a power-law model based on the fracture radius, and the fracture aperture is related to the fracture size using the cubic law [55]. Under such a formulation, the fracture apertures are uniform on each fracture, but vary among fractures. The network is generated in a cubic domain with sides of length one-kilometer. Dirichlet boundary conditions are imposed on the top (1 MPa) and bottom (2 MPa) of the domain to create a pressure gradient aligned with the vertical axis, and no-flow boundary conditions are enforced along lateral boundaries.

A sample realization of the Forsmark DFN is shown in Fig. 5. The realization contains 4,934 fractures and the computational mesh consists of 7,731,299 cells, here the minimal length scale h was set to 2 meters. There are significantly more small, low permeability, fractures than larger fractures due to the powerlaw distribution used to generate the network. The larger fractures act as conduits for flow, connecting numerous small fractures together in the network. Figure 5.a shows the steady state pressure solution. Fractures are colored by pressure, with warmer colors indicating higher values. A selection of two hundred particle trajectories passing through this network are shown in Fig. 5.b. The particles are inserted uniformly along fractures in the inlet plane (bottom), and the movement of the particles is determined by the the local velocity. The only a subset of the fractures is shown along with the trajectories to highlight that particles are attracted toward larger fractures. Because the fracture permeability is based upon the fracture radius, the largest fractures have the highest permeabilities. Although the particles are inserted uniformly, their trajectories cluster together onto larger fractures. This clustering suggests that transport only occurs within a small portion of the fracture network far away from the inlet plane.

Hyman et al. [56], generated a set of statistically independent networks using this experimental setup to study the influence of boundary conditions in Lagrangian transport on asymptotic plume behavior. This was the first work to investigate the significance of injection mode in large, kilometer-scale, three-dimensional discrete fracture networks, which was made possible by DFNWORKS. Due to the size of the domains considered, they were able to confirm for the first time a previously hypothesized scenario that particle clouds injected under resident conditions evolve to behave similarly to clouds injected under flux-weighted conditions. The level of detail obtained using DFNWORKS allowed them to link physical mechanisms to the occurrence of this behavior, and determine that network scale and in-fracture flow channeling lead to the onset of the similarities between the two injection modes.

3.2. Hydraulic Fracturing

Hydraulic fracturing (fracking) has provided access to hydrocarbon trapped in low-permeability media, such as tight shales. The process involves injecting water at high pressures to reactivate existing fractures and also create new fractures to increase permeability of the shale allowing hydrocarbons to be extracted. However, the fundamental physics of why fracking works and its long term ramifications are not well understood. Karra et al., [40] used DFNWORKS to generate a typical production site and simulate production. Using this physics based model, they found good agreement with production field data and determined what physical mechanisms control the decline in the production curve.

A DFN with 383 fractures in a domain of size 200 m × 200 m × 200 m was generated using data from a shale site in the upper Pottsville formation in Alabama; details of the site be found in [57]; here the minimal length scale h was set to 1 meter. A horizontal well is placed at the center of the domain along with six equally spaced fractures that represent hydraulically generated fractures that are perpendicular to the well. Two families of fractures that are generally parallel and perpendicular to the horizontal well are used, with orientations based on the Rose diagram in Figure 6 of [57]. Operational values for permeability, pressure, and porosity were used for realism. For pressure boundary conditions, a pressure of 21 MPa is applied at four boundaries parallel to the well and a bottom hole pressure of 17 MPa is set in the well region. No flow boundary conditions are set on the faces perpendicular to the horizontal well.

Figure 6.a shows the steady-state pressure solution in the DFN and the particle based pathlines obtained using this solution are provided in Fig. 6.b. Using these pathlines, the rate of mass arrival at the horizontal well is then evaluated to generate the production curve shown in Fig. 6.c. For the production curve in Fig. 6.c 100,000 particles are seeded at randomly distributed locations within the DFN. For clarity, only 100 particle trajectories are shown in Fig. 6.b. Matrix diffusion was omitted in this simulation to isolate the effects of advective transport through the DFN. This production curve is calibrated to the Haynesville production data from the report of Moniz et al., [58] by matching the peak production rate to obtain the appropriate amount of volume for each packet. Typical production curves from Barnett, Haynesville and Marcellus shale formations (cf. [58], figure 2.15a), exhibit the same behavior of an initial peak followed by a steep decline that is followed by a long tail. As seen in Fig. 6.c, the advective transport in the DFN leads to a sharp rise initially followed by a steep decline which is then followed by a long tail, indicating that fracture flow causes the initial peak and decline of the production curve. More specifically, particles closest to the well travel quickly to the well causing the initial rise in the production curve while the particles farther away have to travel through complex pathways in the fracture network causing the sharp decline and long tail. However, Karra et al., [40] observe much lower production rates in the tail, suggesting that other mechanisms contribute to the tail.

3.3. CO_2 Sequestration

dfnWorks provides the framework necessary to perform multiphase simulations (such as flow and reactive transport) of these at the reservoir scale. A particular application, highlighted here, is sequestering CO_2 from anthropogenic sources and disposing it in geological formations such as deep saline aquifers and abandoned oil fields. Geological CO_2 sequestration is one of the principal methods under consideration to reduce carbon footprint in the atmosphere due to fossil fuels [59, 60]. For safe and sustainable long-term storage of CO_2 and to prevent leaks through existing faults and fractured rock (along with the ones created during the injection process), understanding the complex physical and chemical interactions between CO_2, water (or brine) and fractured rock, is vital. dfnWorks capability to study multiphase flow in a DFN can be used to study potential CO_2 migration through cap-rock, a potential risk associated with proposed subsurface storage of CO_2 in saline aquifers or depleted reservoirs. Moreover, using the reactive transport capabilities of pflotran coupled with cell-based transmissivity of the DFN allows one to study dynamically changing permeability fields with mineral precipitation and dissolution due to CO_2-water interaction with rock.

Figure 7 shows the temporal evolution of a multiphase CO_2-water flow simulation within a DFN obtained using dfnWorks. The size of the domain is a cubic meter containing 24 fractures; here the minimal length scale h was set to 0.05 meters. Initial hydrostatic pressure conditions, 20 MPa at the bottom of the domain, and isothermal temperature conditions, 50 C throughout the domain, are applied. The domain initially fully saturated with water, and supercritical CO_2 is injected at the bottom of the domain at a rate of 0.01 kg/s for 10 hours. The subfigures in Fig. 7 show the transient dynamics in the system, where color indicates supercritical CO_2 saturation; warmer colors showing higher saturation. The top left subfigures shows the initial state of the system, fully saturated with water. There is an initial flush through the system during the first hour of the simulation (top row of subfigures) as CO_2 is injected into the DFN from the bottom of the domain to displace the water. As time progresses more water is pushed out the top of the domain and CO_2 starts to fill the entire network. After one hour, the rate of displacement decreases (bottom row of subfigures) and the network is partially saturated with both water and supercritical CO_2.

4. Summary and Remarks

DFNWORKS is a parallelized computational suite for simulating single/multiphase flow and transport in stochastically generated three-dimensional discrete fracture networks. We have described the pillars of DFNWORKS (DFNGEN, DFNFLOW, DFNTRANS) and provided several applications of the work flow to demonstrate its utility. DFNWORKS is open source, released under license LA-CC #14−091 and can be obtained by contacting dfnworks@lanl.gov The three applications we presented highlight various aspects of DFNWORKS including its ability to generate networks in accordance with geological data, match transport properties to data, and simulate multiphase flow. Each of the phases of DFNWORKS scales differently depending on number of fractures, CPUs, and particles. Preliminary results show that the network generation scales like $N^{1.5}$, where N is the number of fractures [17]. Recall that this in the only piece of the workflow that is implicitly serial and further advancements could be made. Details about how PFLOTRAN scales with number of control volumes and number of CPUs can be found in [2]. When run on a single CPU, DFNTRANS scales roughly linearly with the number of particles.

We focused on the basic elements of the computational suite to provide a foundational understanding of the workflow. However, there are several extensions of the method that, while not being central to the workflow, increase its utility. These include: (i) in-fracture aperture/transmissivity variability; (ii) networks composed of multiple regions with variable fracture densities; (iii) mapping a DFN into a continuum for the inclusion of fracture-matrix interaction; (iv) multicomponent reactive transport; (v) integration uncertainty quantification and sensitive analysis suites such as MADS (Model Analysis and Decision Support) [61]; (vi) inclusion of large deterministic fractures for site specific studies; (vii) solute transport with sorption and matrix interactions, e.g., trajectories generated by DFNWORKS may be used as input to the MARFA computer code [62]; (viii) The methods of Srinivasan and Lipnikov [63] could be used to generate quasi-optimal accuracy-preserving velocity field as opposed to those adopted here.

Acknowledgements

The various tools under DFNWORKS and the overall workflow has been developed through the support of various funding programs including LANL's DR research project # 20140002DR, U.S. Department of Energy Strategic Center for Natural Gas and Oil project on 'Fundamentals of Unconventional Reservoirs' and the U.S. Department of Energy Used Fuel Disposal campaign. Jeffrey Hyman acknowledges the support of the Center for Nonlinear Studies at Los Alamos National Laboratory through Grant # DE-AC52-06NA25396. Satish Karra thanks Glenn Hammond and Gautam Bisht for their help with explicit unstructured grid implementation and corresponding I/O in PFLOTRAN.

References

[1] S. Neuman, Trends, prospects and challenges in quantifying flow and transport through fractured rocks, Hydrogeology Journal 13 (2005) 124–147.
[2] P. Lichtner, G. Hammond, C. Lu, S. Karra, G. Bisht, B. Andre, R. Mills, J. Kumar, PFLOTRAN User Manual: A Massively Parallel Reactive Flow and Transport Model for Describing Surface and Subsurface Processes, Technical Report Report No.: LA-UR-15-20403, Los Alamos National Laboratory, 2015.
[3] M. Cacas, E. Ledoux, G. d. Marsily, B. Tillie, A. Barbreau, E. Durand, B. Feuga, P. Peaudecerf, Modeling fracture flow with a stochastic discrete fracture network: Calibration and validation: 1. the flow model, Water Resources Research 26 (1990) 479–489.
[4] S. Berrone, S. Pieraccini, S. Scialò, A PDE-constrained optimization formulation for discrete fracture network flows, SIAM Journal of Scientific Computing 35 (2013) B487–B510.
[5] J.-R. de Dreuzy, C. Darcel, P. Davy, O. Bour, Influence of spatial correlation of fracture centers on the permeability of two-dimensional fracture networks following a power law length distribution, Water Resources Research 40 (2004).
[6] J.-R. Dreuzy, Y. Méheust, G. Pichot, Influence of fracture scale heterogeneity on the flow properties of three-dimensional discrete fracture networks, Journal of Geophysical Research - Solid Earth 117 (2012).
[7] W. Dershowitz, FracMan version 7.4-Interactive discrete feature data analysis, geometric modeling, and exploration simulation: User documentation, http://fracman.golder.com/, 2014.
[8] J. Erhel, J.-R. De Dreuzy, B. Poirriez, Flow simulation in three-dimensional discrete fracture networks, SIAM Journal of Scientific Computing 31 (2009) 2688–2705.

[9] H. Mustapha, K. Mustapha, A new approach to simulating flow in discrete fracture networks with an optimized mesh, SIAM Journal of Scientific Computing 29 (2007) 1439.

[10] G. Pichot, J. Erhel, J. De Dreuzy, A mixed hybrid mortar method for solving flow in discrete fracture networks, Applicable Analysis 89 (2010) 1629–1643.

[11] G. Pichot, J. Erhel, J. de Dreuzy, A generalized mixed hybrid mortar method for solving flow in stochastic discrete fracture networks, SIAM Journal of Scientific Computing 34 (2012) B86–B105.

[12] C. Xu, P. Dowd, K. Mardia, R. Fowell, A connectivity index for discrete fracture networks, Mathematical Geology 38 (2006) 611–634.

[13] W. Dershowitz, C. Fidelibus, Derivation of equivalent pipe network analogues for three-dimensional discrete fracture networks by the boundary element method, Water Resources Research 35 (1999) 2685–2691.

[14] S. Joyce, L. Hartley, D. Applegate, J. Hoek, P. Jackson, Multi-scale groundwater flow modeling during temperate climate conditions for the safety assessment of the proposed high-level nuclear waste repository site at Forsmark, Sweden, Hydrogeology Journal 22 (2014) 1233–1249.

[15] S. L. Painter, C. W. Gable, S. Kelkar, Pathline tracing on fully unstructured control-volume grids, Computational Geosciences 16 (2012) 1125–1134.

[16] S. Painter, V. Cvetkovic, J. Mancillas, O. Pensado, Time domain particle tracking methods for simulating transport with retention and first-order transformation, Water Resources Research 44 (2008).

[17] J. D. Hyman, C. W. Gable, S. L. Painter, N. Makedonska, Conforming Delaunay triangulation of stochastically generated three dimensional discrete fracture networks: A feature rejection algorithm for meshing strategy, SIAM Journal of Scientific Computing 36 (2014) A1871–A1894.

[18] LaGriT, Los Alamos Grid Toolbox, (LaGriT) Los Alamos National Laboratory, http://lagrit.lanl.gov, 2013.

[19] G. Zyvoloski, FEHM: A control volume finite element code for simulating subsurface multi-phase multi-fluid heat and mass transfer, Los Alamos Unclassified Report LA-UR-07-3359 (2007).

[20] K. Pruess, C. M. Oldenburg, G. J. Moridis, TOUGH2 user's guide version 2 (1999).

[21] T. Kalbacher, R. Mettier, C. McDermott, W. Wang, G. Kosakowski, T. Taniguchi, O. Kolditz, Geometric modelling and object-oriented software concepts applied to a heterogeneous fractured network from the grimsel rock laboratory, Computational Geosciences 11 (2007) 9–26.

[22] H. Mustapha, R. Dimitrakopoulos, Discretizing two-dimensional complex fractured fields for incompressible two-phase flow, International Journal for Numerical Methods in Fluids 65 (2011) 764–780.

[23] H. Mustapha, R. Dimitrakopoulos, T. Graf, A. Firoozabadi, An efficient method for discretizing 3D fractured media for subsurface flow and transport simulations, International Journal for Numerical Methods in Fluids 67 (2011) 651–670.

[24] J. Maryška, O. Severỳn, M. Vohralík, Numerical simulation of fracture flow with a mixed-hybrid FEM stochastic discrete fracture network model, Computational Geosciences 8 (2005) 217–234.

[25] M. F. Benedetto, S. Berrone, S. Pieraccini, S. Scialò, The virtual element method for discrete fracture network simulations, Computer Methods in Applied Mechanics and Engineering 280 (2014) 135–156.

[26] J. Ruppert, A Delaunay refinement algorithm for quality 2-dimensional mesh generation, Journal of Algorithms 18 (1995) 548–585.

[27] D. Mount, C. Gable, A point-placement strategy for conforming Delaunay tetrahedralization, International Journal of Computational Geometry & Applications 11 (2001) 669–682.

[28] G. A. Zyvoloski, V. V. Vesselinov, An investigation of numerical grid effects in parameter estimation, Groundwater 44 (2006) 814–825.

[29] R. T. Mills, C. Lu, P. C. Lichtner, G. E. Hammond, Simulating subsurface flow and transport on ultrascale computers using PFLOTRAN, in: Journal of Physics: Conference Series, volume 78, IOP Publishing, 2007, p. 012051.

[30] G. E. Hammond, P. C. Lichtner, C. Lu, R. T. Mills, PFLOTRAN: reactive flow & transport code for use on laptops to leadership-class supercomputers, Groundwater Reactive Transport Models (2012) 142–160.

[31] G. E. Hammond, P. C. Lichtner, R. T. Mills, Evaluating the performance of parallel subsurface simulators: An illustrative example with PFLOTRAN, Water Resources Research 50 (2014) 208–228.

[32] S. Balay, S. Abhyankar, M. Adams, J. Brown, P. Brune, K. Buschelman, V. Eijkhout, W. Gropp, D. Kaushik, M. Knepley, et al., PETSc users manual revision 3.5 (2014).

[33] H. Childs, E. Brugger, K. Bonnell, J. Meredith, M. Miller, B. Whitlock, N. Max, A contract based system for large data visualization, in: Visualization, 2005. VIS 05. IEEE, IEEE, 2005, pp. 191–198.

[34] J. Ahrens, B. Geveci, C. Law, 36 paraview: An end-user tool for large-data visualization, The Visualization Handbook (2005) 717.

[35] N. Makedonska, S. L. Painter, C. W. Bui, Q. M. Gable, S. Karra, Particle tracking approach for transport in three-dimensional discrete fracture networks, Computation Geosciences (under review) (2015).

[36] P. Lichtner, S. Karra, Modeling multiscale-multiphase-multicomponent reactive flows in porous media: Application to co2 sequestration and enhanced geothermal energy using PFLOTRAN, in: Al-Khoury, R., Bundschuh, J. (eds.) Computational Models for CO2 Geo-sequestration & Compressed Air Energy Storage (http://www.crcnetbase.com/doi/pdfplus/10), CRC Press, 2014, pp. 81–136.

[37] C. Lu, P. C. Lichtner, High resolution numerical investigation on the effect of convective instability on long term co2 storage in saline aquifers, in: Journal of Physics: Conference Series, volume 78, IOP Publishing, 2007, p. 012042.

[38] G. E. Hammond, P. C. Lichtner, Field-scale model for the natural attenuation of uranium at the Hanford 300 area using high-performance computing, Water Resources Research 46 (2010).

[39] R. S. Middleton, J. W. Carey, R. P. Currier, J. D. Hyman, Q. Kang, S. Karra, J. Jiménez-Martínez, M. L. Porter, H. S. Viswanathan, Shale gas and non-aqueous fracturing fluids: Opportunities and challenges for supercritical CO_2, Applied

Energy 147 (2015) 500–509.
[40] S. Karra, N. Makedonska, H. Viswanathan, S. Painter, J. Hyman, Effect of advective flow in fractures and matrix diffusion on natural gas production, Water Resources Research (under review) (2015).
[41] S. Karra, G. Bisht, P. C. Lichtner, G. E. Hammond, Coupling geomechanics with flow and reactive transport in PFLOTRAN for subsurface applications, in: AGU Fall Meeting Abstracts, volume 1, 2013, p. 1106.
[42] S. Karra, S. L. Painter, P. C. Lichtner, Three-phase numerical model for subsurface hydrology in permafrost-affected regions (PFLOTRAN-ICE v1. 0), The Cryosphere 8 (2014) 1935–1950.
[43] R. T. Mills, G. Bisht, S. Karra, F. M. Hoffman, G. E. Hammond, J. Kumar, S. L. Painter, P. E. Thornton, P. C. Lichtner, Progress towards coupled simulation of surface/subsurface hydrologic processes and terrestrial ecosystem dynamics using the community models PFLOTRAN and CLM, in: AGU Fall Meeting Abstracts, volume 1, 2012, p. 1426.
[44] D. Elsworth, A hybrid boundary element-finite element analysis procedure for fluid flow simulation in fractured rock masses, International Journal for Numerical and Analytical Methods in Geomechanics 10 (1986) 569–584.
[45] L. Hartley, I. Cox, D. Holton, F. Hunter, S. Joyce, B. Gylling, M. Lindgren, Groundwater flow and radionuclide transport modelling using CONNECTFLOW in support of the SR Can assessment, Technical Report SKB Rapport R-04-61, Swedish Nuclear Fuel and Waste Management Co., Stockholm (Sweden), 2004.
[46] A. Frampton, V. Cvetkovic, Numerical and analytical modeling of advective travel times in realistic three-dimensional fracture networks, Water Resour. Res. 47 (2011).
[47] J. Geier, Investigation of discrete-fracture network conceptual model uncertainty at Forsmark, Technical Report, Swedish Radiation Safety Authority, Stockholm (Sweden), 2011.
[48] N. Outters, D. Shuttle, Sensitivity analysis of a discrete fracture network model for performance assessment of Aberg, Technical Report, Swedish Nuclear Fuel and Waste Management Co., Stockholm (Sweden), 2000.
[49] H. S. M. Coxeter, H. S. M. Coxeter, H. S. M. Coxeter, H. S. M. Coxeter, Introduction to geometry, volume 6, Wiley New York, 1969.
[50] L. C. Hull, K. N. Koslow, Streamline routing through fracture junctions, Water Resources Research 22 (1986) 1731–1734.
[51] Y.-J. Park, J.-R. De Dreuzy, K.-K. Lee, B. Berkowitz, Transport and intersection mixing in random fracture networks with power law length distributions, Water Resources Research 37 (2001) 2493–2501.
[52] Y.-J. Park, K.-K. Lee, G. Kosakowski, B. Berkowitz, Transport behavior in three-dimensional fracture intersections, Water resources research 39 (2003).
[53] SKB, Long-term safety for the final repository for spent nuclear fuel at Forsmark. Main report of the SR-Site project., Technical Report SKB TR-11-01, Swedish Nuclear Fuel and Waste Management Co., Stockholm (Sweden), 2011.
[54] L. Hartley, S. Joyce, Approaches and algorithms for groundwater flow modeling in support of site investigations and safety assessment of the Forsmark site, Sweden, Journal of Hydrology 500 (2013) 200–216.
[55] P. M. Adler, J.-F. Thovert, V. V. Mourzenko, Fractured Porous Media, Oxford University Press, 2012.
[56] J. Hyman, S. L. Painter, H. S. Viswanathan, N. Makedonska, S. Karra, Influence of injection mode on transport properties in kilometer-scale three-dimensional discrete fracture networks, Water Resources Research (Accepted) (2015).
[57] G. Jin, J. Pashin, J. Payton, Application of discrete fracture network models to coalbed methane reservoirs of the Black Warrior basin: Tuscaloosa, Alabama, University of Alabama College of Continuing Studies, in: 2003 International Coalbed Methane Symposium Proceedings, Paper, volume 321, 2003, p. 13.
[58] E. J. Moniz, H. D. Jacoby, A. Meggs, R. Armstrong, D. Cohn, S. Connors, J. Deutch, Q. Ejaz, J. Hezir, G. Kaufman, The future of natural gas, Cambridge, MA: Massachusetts Institute of Technology (2011).
[59] S. Bachu, Sequestration of CO_2 in geological media in response to climate change: road map for site selection using the transform of the geological space into the CO_2 phase space, Energy Conversion and management 43 (2002) 87–102.
[60] S. Pacala, R. Socolow, Stabilization wedges: solving the climate problem for the next 50 years with current technologies, science 305 (2004) 968–972.
[61] V. Vesselinov, D. Harp, Model analysis and decision support (mads) for comple physics models, in: XIX International conference on water resources-CMWR, 2012.
[62] S. Painter, J. Mancillas, P. Oy, MARFA user's manual: migration analysis of radionuclides in the far field, Posiva Working Report 1 (2013).
[63] G. Srinivasan, K. Lipnikov, On the reconstruction of Darcy velocity in finite-volume methods, Transport Porous Med. 96 (2013) 337–351.

List of Figures

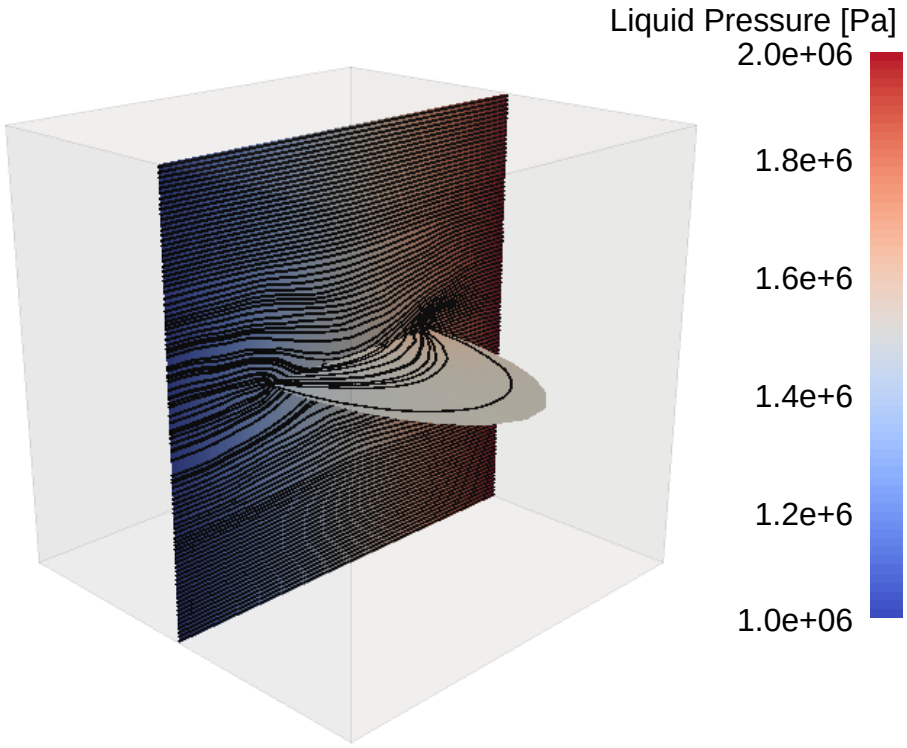


Figure 1: Two fracture DFN demonstrating the utility of the dfnWorks suite. Fractures are colored by the steady-state pressure solution, the gradient is aligned with the x-axis, and black lines are particle trajectories. The non-uniformity of the trajectories results in a distribution of particle travel times that exhibits longitudinal dispersion. Using the simplified pipe-network approximations, the breakthrough curve would be a step function with no dispersion or tail, which will lead to incorrect upscaled models for transport.

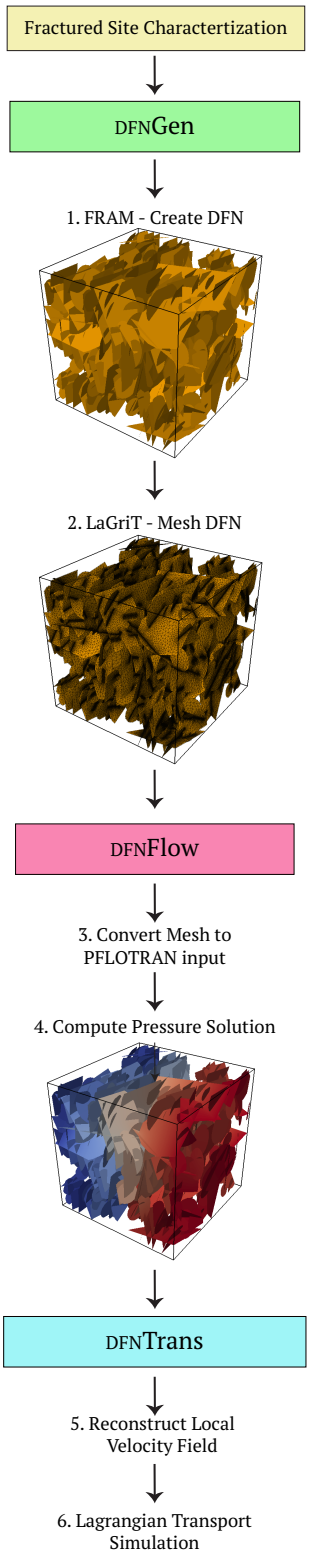


Figure 2: DFNWORKS Workflow. From top: The input for DFNWORKS is a fractured site characterization that provides distributions of fracture orientations, radius, and spatial locations. DFNGEN: 1) FRAM - Create DFN: Using the fractured site characterization that networks are constructed using the feature rejection algorithm for meshing. 2) LAGRIT - Mesh DFN: The LAGRIT meshing tool box is used to create a conforming Delaunay triangulation of the network. DFNFLOW: 3) Convert Mesh to PFLOTRAN input: Control volume information is formatted for PFLOTRAN. 4) Compute Pressure Solution: The steady-state pressure solution in the DFN is obtained using PFLOTRAN. DFNTRANS: 5) Reconstruct Local Velocity Field: Darcy fluxes obtained using DFNFLOW are used to reconstruct the local velocity field, which is used for particle tracking on the DFN. 6) Lagrangian Transport Simulation: An extension of the WALKABOUT method is used to determine pathlines through the network and simulate transport.

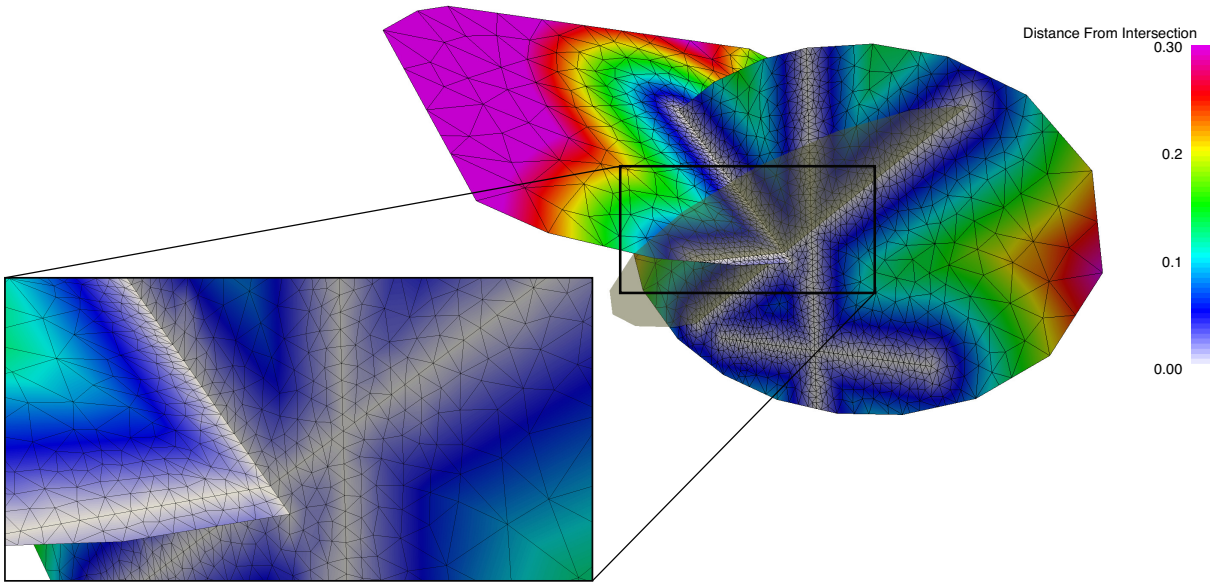


Figure 3: Three intersecting fractures show the intersecting conforming Delaunay triangulations. Two of the fractures are colored by distance from lines of intersections (traces) between fractures, and the other is semi-transparent. The mesh is optionally coarsened away from intersections with pressure gradients will be lower. The inclusion of the semi-transparent fracture illustrates how FRAM creates a mesh that adheres to multiple intersections on the surface of a single fractures. Two additional fractures intersect the elliptical fracture, and intersect one another on the surface of that fracture, as shown by the intersecting white colored regions. The inset shows that the Delaunay mesh conforms all of these lines of intersection.

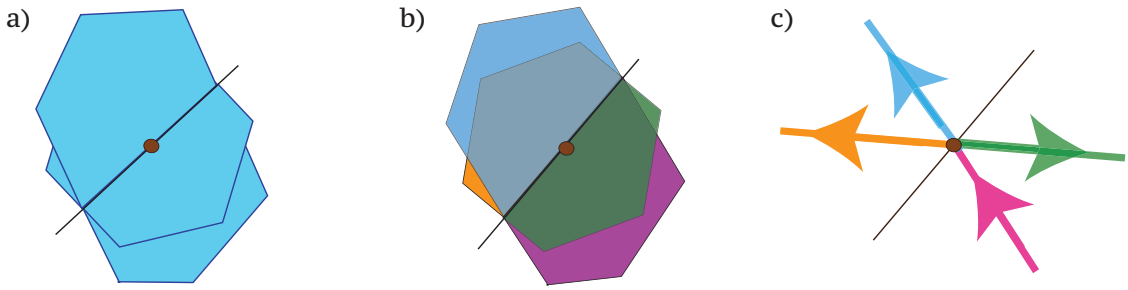
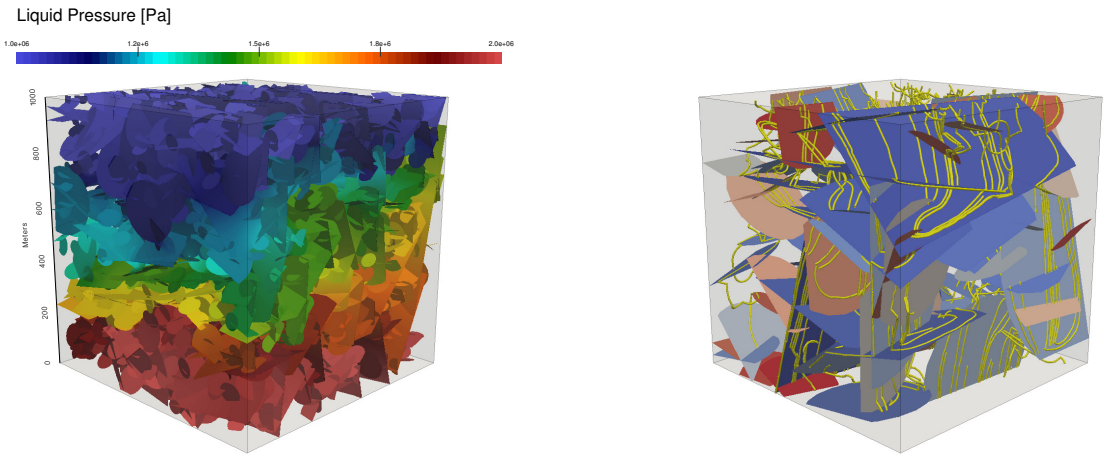


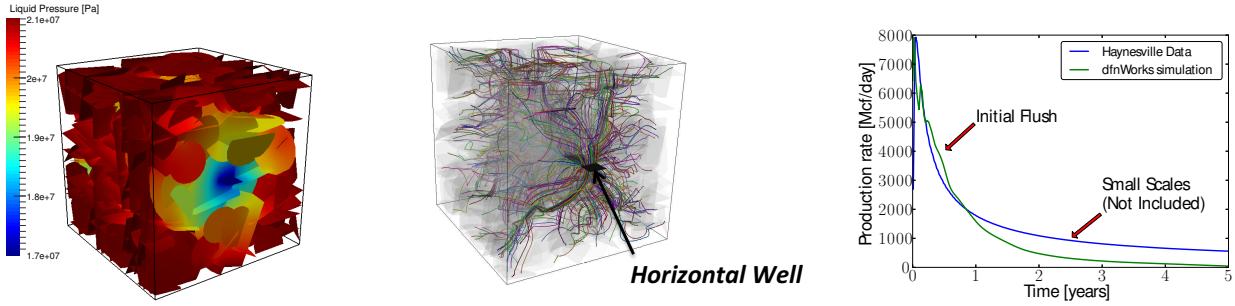
Figure 4: Illustration of the technique used to reconstruct flow velocities along the line of intersection between two fractures. a) A control volume along a line of intersection between two fractures. Fluxes are defined on faces of the control volume cells. b) The control volume is spilt into four sub-polygons using the line of intersection. The green and orange sub-polygons are on one fracture and the blue and purple are on another. c) The flow velocities are reconstructed on each of the split polygons, here arrows indicate inflow/outflow. In this example, the flux into the control volume occurs through the purple sub-volume and outflow occurs through the remaining sub-polygons.



(a) Pressure Solution

(b) Pathlines

Figure 5: Modified from Hyman et al. [56]. a) A DFN realization based upon the fractured granite at the Forsmark, Sweden site in a 1 km^3 domain. The model consist of multiple sets of circular fractures whose orientations follow a Fisher distribution. The fracture diameters in each fracture set follow a truncated power-law distribution with lower cutoff at 15 meters. Fractures are colored by pressure. Dirichlet boundary conditions are imposed on the top (1 MPa) and bottom (2 MPa) of the domain to create a pressure gradient in the z direction, and no-flow boundary conditions are imposed along lateral boundaries. b) Two hundred particle trajectories within the DFN shown in Fig. 5. Particles are inserted uniformly along fractures on the bottom of the domain. Trajectories are overlaid on a subset of the network shown in Fig. 5(a) to show how particles are drawn to larger fractures, which have higher permeabilities and offer less resistance than smaller fractures. Although particles are inserted uniformly along fractures in the inlet plane, they cluster on larger fractures as they exit the domain. This clustering suggests that transport only occurs within a small portion of the fracture network far away from the inlet plane.



(a) Pressure solution (b) Particle pathways (c) Production curve

Figure 6: Modified from Karra et al. [40] (a) Discrete fracture network used in this work and the pressure solution computed using PFLOTRAN on the DFN. (b) Transport pathlines of gas packets represented by 1000 dynamically inert and indivisible tracer parcels (particles) traveling to the horizontal well. (c) The resulting production curve produced using the pathlines in (b). The maximum value of the production curve has been calibrated to the Haynesville production data from Moniz's report [58].

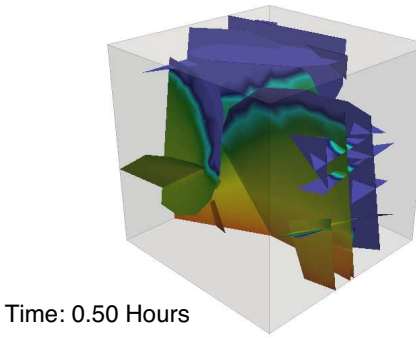


Figure 7: Temporal evolution of supercritical CO_2 displacing water in a meter cube DFN containing 24 fractures. The DFN is initially fully saturated with water, (top left Time 0 hours) and supercritical CO_2 is slowly injected into the system from the bottom of the domain to displace the water for a total time of 10 hours. There is an initial flush through the system during the first hour of the simulation, and then the rate of displacement decreases.